REMARKS

Claims 35-42 and 44-55 are all the claims presently pending in the Application. Claim 43 is cancelled. New claim 55 has been added.

It is noted that any claim amendments are made to merely clarify the language of each claim, and not for distinguishing the invention over the prior art, narrowing the claims or for any statutory requirements of patentability. It is further noted that, notwithstanding any claim amendments made herein, Applicant's intent is to encompass equivalents of all claim elements, even if amended herein or later during prosecution.

Claims 35-54 stand rejected under 35 USC 103(a) as being unpatentable over Robertazzi et al (U.S. 6,370,560) in view of Maher et al (U.S. 6,301,603). The rejections are respectfully traversed in view of the following discussion.

THE DOUBLE PATENTING REJECTION

The Examiner provisionally rejected claims 35, 46, and 53 under the doctrine of obviousness-type double patenting as being unpatentable over co-pending applications 09/871,775 and 09/871,776. Applicant has notified Examiner Vo who was examining 09/871,775 that Applicant will abandon that application and will pursue the claims in the present application.

First, claim 1 of the '776 application was canceled during prosecution, so the provisional rejection to that claim is moot. However, claim 9 of that application will be

discussed below to prevent any further allegations of double patenting rejections. The final version of the amended claims in the '776 application are not obvious over the amended claims in the current application. The claims in the '776 application are means-for claims directed to different elements and functions of an apparatus that includes an allocating means for "functions waiting in a queue" using a specific priority scheme, with a specific load determining determining means, for "j instances of ...k signal processing functions," "assigning a value of either zero or one to a multiplicand," multiplying the estimate amount of the processing resouce ...," "summing together," and numerous other different limitations in that application that do not render obvious the present claims. A copy of the finally-amended claim 9 from the '776 (the only independent claim currently pending) is as follows:

> 9. An apparatus for allocating processing resources of a signal processor to signal processing functions, which are associated with inputted signals, comprising:
>
> a capacity determining means for determining an amount of the processor resources available to be assigned to the signal processing functions;
>
> a load determining means for determining an estimate of an amount of the processing resources needed for each of the signal processing functions waiting in a queue [[to]] to be executed;
>
> an allocating means, which receives information from said capacity determining means and said load determining means, for allocating the available processing resources to signal processing functions waiting in the queue to be executed, based on a hierarchical priority scheme, wherein
>
> said load determining means calculates a product, for each of j instances, corresponding to each of said inputted signals, where j = 1 to m, and for each of k signal processing functions associated with each of said j instances, where k = 1 to N, obtained by:
>
> (a) estimating an amount of processing resource needed to support the execution of the j^{th} instance of the k^{th} signal

processing function;

(b) assigning a value of either zero or one to a multiplicand associated with the j^{th} instance of the k^{th} signal processing function; and

(c) multiplying the estimated amount of processing resource needed to support the execution of the j^{th} instance of the k^{th} signal processing function by its associated multiplicand and assigning the result to the product associated with the j^{th} instance of the k^{th} signal processing function; and

said load determining means calculates a sub-total sum, for each of the j instances, obtained by:

(d) summing together the products associated with each of the k signal processing functions associated with each of the j instances; and

(e) adding another estimated amount of the processing resource needed to support background processing associated with each of the j instances to the sum of products of each of the k signal processing functions associated with each of the j instances and assigning the result to the sub-total for each of the j instances.

Due to the numerous differences between the '776 claim 9 and present application claims, Applicant respectfully requests the Examiner reconsider and withdraw the provisional double-patenting rejection.

THE 35 USC 112, SECOND PARAGRAPH REJECTION

Claims 35-45 stand rejected for the term "anticipating the exceedance" (line 7) of claim 35. This term has been removed from the claims by the above-listed claim amendments and is therefore moot. The Examiner is respectfully requested to reconsider and withdraw the rejection.

THE PRIOR ART REJECTION

In the Sept. 30, 2005 amendment, the Examiner again cited Robertazzi against the present claims. However, the Examiner noted in Section 9 that "Robertazzi does not specifically teach the processor being a signal processor for providing signal processing" and therefore added Maher to form a 103 rejection. However, Applicant has maintained throughout this prosecution that Robertazzi is inapplicable to the claimed invention, and Maher fails to make up for Robertazzi's shortcomings.

Although Applicant made amendments to the claims in prior responses, after a review of all the references cited in this prosecution Applicant asserts that none of the references, alone or in combination, cited thus far in prosecution teach or suggest the claimed invention. Therefore, the present claim amendments are intended to describe the proper scope of the claimed invention.

The problem with Robertazzi, Maher, and the formerly cited references is fundamental and has been explained consistently in all the prior responses from Applicant: the combination of references all use distributed processing to process a fixed load on multiple processors. In other words, when a set of instructions or data must be processed, but the capabilities of the core processor cannot handle the load, the load is divided out onto other separate processors. The techniques of Robertazzi, Maher, and all other distributed processing technology is the problem in the prior art that the present invention overcomes: how to process a load that would traditionally be distributed between different processors all on a single processor. The present invention accomplishes this goal in part using software processes that are not fixed, in

other words they are "manageable" functions of an algorithm as recited in claim 1. The present invention uses "a single processor" that processes "manageable functions," as recited in claim 1. This is a fundamental flaw with Robertazzi and Maher and the reason why, even from the basic view of the technology in those references, shows that they cannot teach or suggest the claimed invention:

> The distributed processor just allocated will then change its availability status to "busy" to indicate it is not currently available for other loads. Then, in later steps, subsequent available distributed processors with the lowest monetary cost will be allocated with the remaining segments of the load or task. (Robertazzi, C. 8, L. 45-50)
>
> A need remains in the art for apparatus and methods which allow music synthesis and audio effects processing to dynamically scale from a default processor to one or more additional processors which may not be of the same type--for example from a DSP to the host CPU--in a manner which permits the audio system to support more tasks as the need arises. (Maher, C. 2, L. 15-24)
>
> Apparatus according to the present invention dynamically allocates audio processing load between at least two heterogeneous audio processors ... (Maher, C. 3, L. 21-23) (emphasis added)

The combination of references clearly disclose multiple processors to handle a fixed, distributed processing load. During the entire prosecution, Applicant has stated consistently that this these aspects are diametrically opposite the disclosure, claims, and result of the claimed invention. Neither Robertazzi, Maher, nor any of the cited art in the prosecution record disclose or teach technology that can be compared to the claimed invention. In the claimed invention, only one processor is used to handle the load, where the load is divisible into manageable software functions according to

different criteria. The claimed invention uses no extra, additional, multiple, or distributed processors and does not manage entirely fixed loads. Therefore, the present invention is a novel technique of processing resources that allows a single processor to handle loads that would otherwise be handled by multiple processors using some type of distributed processing scheme. Claim 1 states that the method is for management of processing resources "in a single processor when inadequate processing resources are available in the processor." In the cited combination of references, as quoted above, such a load would be sent to additional processors. In contrast, when there are a "plurality of functions of an adaptive algorithm," and "an execution of each function is manageable" the method allocates processing resources of the single processor according to an estimate use and achieved performance of the function, and then controls "the execution of each function according to the allocation of the processing resources." Thus, all the allocation and execution of the adaptive algorithm is performed in the single processor.

Such features are not found anywhere either Robertazzi, Maher, or their combination. In Maher, the goal is to find a way to scale from a default processor onto multiple processors, which is the opposite technique of the claimed invention. (see Col. 2, L. 20). In Robertazzi, the goal is to use many cheap processors for "allocation of divisible load jobs among a plurality of distributed processor platforms based on their resource utilization costs." (Col. 2, L. 52-54). As explained in all the prior responses, Robertazzi's divisible load under the divisible load theory, and is a fixed load that can be divided and distributed and therefore delayed in processing. (Col. 1, L. 50-60). One

skilled in the art knows that these are not the manageable functions of an adaptive algorithm that are processed by the claimed invention. Those disclosures have not choice but to use many additional processors when the first processor becomes full or "busy" and overloaded with processing requests or to simply not process the loads at all, as quoted above.

In regard to claim 36, contrary to the Examiner's allegations, Robertazzi does not teach or suggest "allocating the processing resources among each function based on an environmental input." The cites to Robertazzi discuss distributing a fixed load to multiple processors according to a financial input of how cheap the processing cost is. This is not processing on a single processor, a financial input is not a processing environmental input, and processing a manageable function of an adaptive algorithm is not disclosed by Robertazzi.

In regard to claim 37, Robertazzi does not teach or suggest allocating allocating processing resources on a single processor to a function of an adaptive algorithm "based on the estimated use of the processing resources by each function and the achieved performance of each function according to a hierarchical priority scheme." Robertazzi controls the divisible load to a time constraint and cost of processing on multiple processors. (Col. 8, L. 15-30) A time constraint for multiple processors of fixed loads does not teach or suggest processing on a single processor, and processing a manageable function of an adaptive algorithm, and allocating to functions according to a hierarchical priority scheme.

In regard to claim 38, Robertazzi does not teach or suggest allocating

processing resources on a single processor to a function of an adaptive algorithm "according to a round-robin priority scheme." Robertazzi controls the divisible load to a time constraint and cost of processing on multiple processors. (Col. 8, L. 15-30) A time constraint for multiple processors of fixed loads does not teach or suggest processing on a single processor and processing a manageable function of an adaptive algorithm according to a hierarchical priority scheme. Robertazzi discloses distributing divisible load jobs to multiple processor to save monetary costs (Col. 2, L 52-55), which is not comparable to a round-robin scheme.

In regard to claim 39, Robertazzi does not teach or suggest allocating processing resources on a single processor to a function of an adaptive algorithm by "removing a portion of the allocated processing resources from each manageable function that can execute using fewer processing resources than were initially allocating during the time period." Robertazzi "reallocates an incremental portion of the load from the most expensive processor platform to the cheaper processor platform." (Col. 11, L. 46-48) Reallocating the same load for processing to save money is not comparable to reducing processing resources to a function that can still execute with few resources than initially allocated. Robertazzi's loads use the same resources whether on a cheap chip or an expensive chip.

In regard to claim 40, does not teach or suggest controlling the execution of each function of an adaptive algorithm on a single processor by "performing re-allocation of fewer of the processing resources to each of the functions that are manageable for performance-degrading execution." Robertazzi discloses a

conventional queueing scheme using a FIFO buffer. (Col. 5, L. 34-60). A FIFO buffer is not comparable to degrading the performance of a software function but still executing the function. Robertazzi does not state or suggest that its loads can be degraded in performance yet still execute.

Regarding claim 41, since Robertazzi has no feature or suggestion for performance-degrading execution of a function of an adaptive algorithm on a single processor as described above, it cannot teach or suggest "re-allocating more of the processing resources to each performance-degraded function when a cumulative usage of said processing resources by the functions fall below said low usage threshold." Reallocating the same load for processing to save money is not comparable to reducing processing resources to a function that can still execute with few resources than initially allocated and the reallocating resources when the total use of resources fall below a threshold. Robertazzi's loads use the same resources whether on a cheap chip or an expensive chip. (Col. 11, L. 46-48)

Regarding claim 42, Robertazzi reallocates the same load for processing to save money. This does not teach or suggest one of enabling each function of an adaptive algorithm on a single processor for executing, disabling to prevent execution, or degrading execution by allocating fewer processing resources if the function is capable of performance-degraded execution. Robertazzi has not such provision nor suggestion for managing adaptive algorithms. Robertazzi's loads are all executed and use the same resources whether on a cheap chip or an expensive chip.(Col. 11, L. 46-48)

Regarding claim 44, Robertazzi allows the user, not the algorithm, to choose how long a process operation will take. Robertazzi then distributes the loads to the cheapest chips to get the job done under the time constraint. (Col. 8) This does not teach or suggest "storing the estimate of maximum required processing resource for execution of each function and a minimum required processing resource for execution of each function for the controlling of the execution of each function." No provision for storage of execution times of an adaptive algorithm function is suggested in the reference.

Regarding claim 45, Robertazzi and Maher both clearly distribute processing to many different alternate processors. This plainly does not teach or suggest controlling the execution of the plurality of functions of the algorithm without executing any function on an additional processor."

Applicant's responses for claims 46-49, 50-52, and 53-54 are incorporated from above according to the corresponding claims recited by the Examiner in paragraphs 21-23.

Should the Examiner find the Application to be other than in condition for allowance, the Examiner may contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any fees associated with this communication to Client's Deposit Account No. 20-0668.

Respectfully submitted,

Kendal M. Sheets, Reg. No. 47,077
Joseph J. Zito, Reg. No. 32,076
Customer No. 23494
(301) 601-5010

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450 on March 30, 2006.

Kendal M. Sheets 3/30/06
Date